Time Synchronization Testing with Roving WLAN Network

Samuel Nyarko

Bachelor's Thesis

Accepted ___.___._____ _____

SAVONIA UNIVERSITY OF APPLIED SCIENCES **THESIS**
Abstract

Field of Study Bachelor of Engineering			
Degree Programme Information Technology			
Author(s) Samuel Nyarko			
Title of Thesis Time Synchronization Testing With Roving WLAN Network			
Date	31October 2011	Pages/Appendices	50
Supervisor(s) Mr. Arto Toppinen, Principal Lecturer			
Client Organization/Partners Savonia University of Applied Sciences			

Abstract

The main objective of this thesis was to synchronize the local clock system of a reference time source and provide the time to other NTP or SNTP clients using the roving Wlan module (Rn-131g-eval) in conjunction with Meinberg NTP Software. Time synchronization in WSN aims to provide a timescale for local clock of nodes in the network.

Wireless Sensor Networks (WSNs) are predominately used in today's networked environment which wirelessly connects small sensor nodes which have the ability of computation, communication and sensing. The sensor nodes collaborate to help in processing vital information/data which are needed for computing and networking applications.

The thesis was carried out with the help of Roving WLAN Network (Rn-131g-eval) and Meinberg NTP Time Server which is a graphical user interface which displays the current status of NTP status and NTP service.

The results elaborate how the clocks were synchronized to an external source for time to be propagated to the clients and wireless ad-hoc network was used to evaluate the working ability of the NTP services with Wireshark for the delay and error to be recorded.

Keywords

Synchronization, NTP, wireless sensor network, ad-hoc, Rn-131g-eval,SNTP

Confidentiality

Public

SAVONIA-AMMATTIKORKEAKOULU TEKNIIKKA KUOPIO
Koulutusohjelma
Tietotekniikan koulutusohjelma

Tekijä

Samuel Nyarko

Työn nimi
Time Synchronization Testing With Roving WLAN Network

Date	31October 2011	Pages/Appendices	50

Työn valvoja

Arto Toppinen

Yritys

Savonia Ammattikorkeakoulu

Tiivistelmä

Työn päätavoite oli synkronisoida paikallinen kellosysteemi levittävästä aikalähteestä ja välittää aika muille NTP tai SNTP asiakkaille, jotka käyttävät kiertävää Wlan moduulia (Rn-131g-eval) yhdistettynä Meinberg NTP Softwareen. Ajan synkronisointi WSN:ssä tähtää verkoston paikallisen noodikellon aikaskaalan välittämiseen.

Wireless Sensor Networks (WSNs) on laajasti käytössä tämän päivän verkostoidussa ympäristössä, joka langattomasti yhdistää pieniä sensorinoodeja, joilla on tietojenkäsittelyn, kommunikaation ja ymmärtämisen kyky. Sensorinoodit työskentelevät yhdessä auttaakseen välttämättömän tiedon prosessoinnissa, jota myöhemmin tarvitaan toisiin tietojenkäsittely- ja verkostosovelluksiin.

Työ suoritettiin Roving WLAN Network (Rn-131g-eval) ja Meinberg NTP Time Serverin avulla, joka on graafinen käyttäjäliitäntä, joka näyttää NTP status ja NTP servicen tämänhetkisen statuksen.

Metodi, joka käsittelee miten kellot synkronisoitiin ulkopuoliseen lähteeseen, jotta aika leviäisi asiakkaille, esitettiin, ja myös langattoman ad-hoc verkoston toteutusta käytettiin NTP services with Wiresharkin työkyvyn arvioinnissa.

Avainsanat

Luottamuksellisuus
julkinen

Acknowledgements

My sincere gratitude goes to the Almighty God for the strength He endowed on me during the difficult moments when I lost my beloved three months three days old son in February 2011.

I thank all my friends and the lecturers that were instrumental for my success in academics from the 2008 to 2012.

My special thanks to Mr. Arto Toppinen for his support and dedication when I was working on this thesis.

February 26, 2012

Kuopio

Samuel Nyarko

Abbreviations

WSN	Wireless Sensor Network
RBS	Reference Broadcasting Synchronization
TPSN	Timing Synchronization Protocol for Sensor Networks
LTS	Lightweight Time Synchronization Protocol
NTP	Network Time Protocol
SNTP	Simple Network Time Protocol
GPS	Global Positioning System
UDP	User Datagram Protocol
UTC	Coordinated Universal Time
WLAN	Wireless Local Area Network
SSID	Service Set Identifier
LBA	Location Based Algorithm
WPA	Wireless Fidelity Protected Access
UART	Universal Asynchronous Receiver/Transmitter
DHCP	Dynamic Host Configuration Protocol
RTC	Real Time Clock
TDMA	Time Division Multiple Access
WLAN	Wireless Local Area Network

Table of Contents

1. INTRODUCTION....8
2. WIRELESS SENSOR NETWORK....9
2.1 Architecture of WSN....9
2.2 Wireless Communication Requirements....9
2.3 Challenges in Wireless Sensor Networks....11
2.3.1 Hardware and operating system for WSN....11
2.3.2 Wireless Radio Communication Characteristics....12
2.3.3 Localization....13
2.3.4 Quality of Service....13
2.3.5 Security....14
2.4 Characteristics of WSNs....15
2.5 Algorithm for Wireless Sensor Networks....15
3. TIME SYNCHRONIZATION....17
3.1 Relevance of Time Synchronization in WSN....17
3.2 Time Synchronization Protocol....17
3.2.1 Reference Broadcasting Synchronization (RBS)....18
3.2.2 Timing Synchronization Protocol for Sensor Networks (TSPN)....19
3.2.3 Lightweight Time Synchronization Protocol (LTS)....20
4. NETWORK TIME PROTOCOL....22
4.1 Components of NTP....22
4.2 Operation of NTP....23
4.2.1 Hierarchical Model....23
5. TOOLS....25
5.1 Meinberg NTP Software....25
5.2 NTP Time Server Monitor....25
5.3 Roving WLAN Module (Rn-131g-eval)....26
6. EXPERIMENT....28
6.1 Hardware Setup....28
6.1.1 Graphical Representation....28
6.1.2 Evaluation Board Setup....28
6.2 Configuring Rn-131g-eval....31

6.2.1 Configuring Module Through UART....32
6.2.2 Associating with an Access Point....33
6.2.3 Configuring Ad-Hoc Mode....34
6.2.4 Configuring Automatic Connections....35
6.2.5 Real Clock Configuration....36
6.3 Configuring NTP....37
6.3.1 Configuring NTP Client....37
6.3.2 Configuring NTP Server....38
6.3.3 Configuring Authentication....39
6.3.4 Managing Effect of Network....40
6.3.5 Configuring Access Control....40
7. RESULTS....42
8. CONCLUSION....44
REFERENCES....45
APPENDICES....49
Appendix 1 Configuration file....49
Appendix 2 Graphical Display of Stratum Value....51
Appendix 3 Graphical Display of Delay....51
Appendix 4 Graphical Display of Polling Interval Delay....52

1. INTRODUCTION

Keeping time accurate and precise on different devices has become a major challenge in today's network environment since a slight change in time can lead an organization into several problems and mishaps. This calls for all the devices to be synchronized to an external reference source which is accurate for time to be distributed to all other devices directly connected.

In recent times, there have been lots of changes in how devices communicate with each other. Communication has taken a different shift from the normal traditional mode of communication. Wireless communication is rapidly gaining momentum in the industry with the ability to provide a high quality information/data exchange with devices they are connected anywhere or in close proximity.

Due to the current ramification in the segment of wireless technologies, it is viable to implement inexpensive, tiny disposal and low power devices for measurement of physical environments like temperature, pressure, humidity etc. The devices that are used to carry out this task are termed as sensor nodes and their deployment is called Wireless Sensor Network (WSN).

In WSNs, the nodes are to sense the environment for an event and communicate them to the base station. To recognize and record the correct event time, the nodes are synchronized among each other with a universal time from a reference clock for time to be distributed to others in the same spectrum. Synchronization in a sensor network provides a common timescale for local clocks of nodes in the network.

In this thesis, the various protocols of Time Synchronization and the challenges in WSNs will be explained with more emphasis placed on Network Time Protocol (NTP). The thesis will be carried out with Meinberg NTP Software and Roving WLAN Network (Rn-131g-eval). Wireshark will be installed to enable packets that travel from one node to another to be recorded with their designated timestamps.

The time obtained from the server connected to the external reference clock will be recorded and compared to ascertain the delays and errors.

2. WIRELESS SENSOR NETWORK

2.1 Architecture of WSN

The best way of divulging into the challenges of building a low-power wireless communication platform is comprehending the intricacies of the system requirements for the communication mechanisms. The main purpose of this chapter is to develop a general architecture that pinpoints the needs of the WSN. It is designed to cooperate with the subsystem that permits for flexibility and application specific optimization of communication protocols while simultaneously obtaining high bandwidth and efficiency. (Sharma 2009, 6)

WSN systems are heterogeneous and complex. Due to this nature, it is imperative to carry out a due diligence on the design to ascertain the necessary parameters. The latter can be solved by technological advancement which enables WSN to make good use of power optimization. How far a WSN algorithm goes in the future depends on factors such as computation, communication and storage cost which operate on different spectrum of the WSN. (Angora, Swati, Richa, Priyanka 2010, 2)

2.2 Wireless Communication Requirements

For communication to take place over a wireless link, it must be necessitated by a protocol which must be instituted from a raw electromagnetic primitive. A transmitter which is responsible for modulating the RF signal is needed for propagating the signal while the receiver performs demodulation and signal processing. Fig 2.1 show wireless communications in WSN

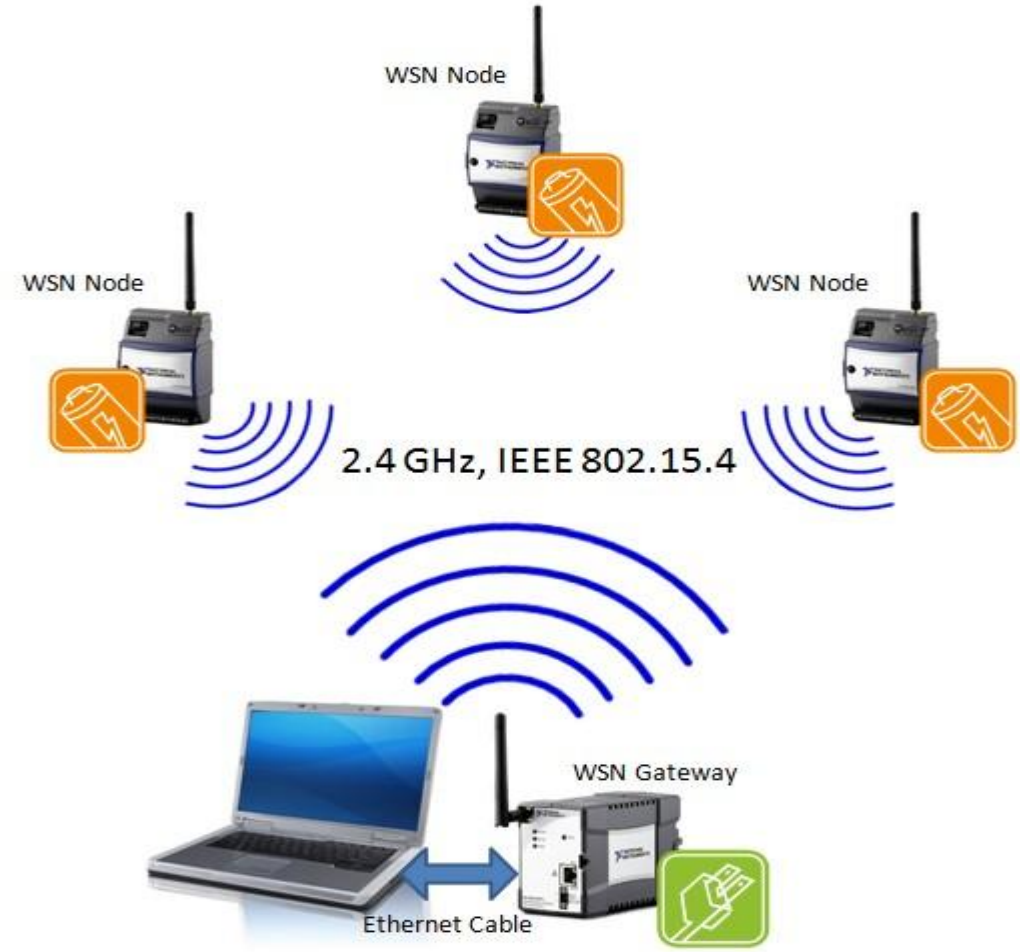


FIGURE 2.1 Wireless Communications in WSN (National Instruments 2010).

The process of communicating over the wireless link involves encoding the data for transmission. The coding schemes are built to raise the efficacy of transmission by avoiding and making amendments to slightly made errors. Transmission begins to take place when the byte is encoded. While the bytes are being transmitted, the remains are at the same time encoded to be transmitted. (Walters, Liang, Shi, Chaudhary 2006. 4)

The propagated symbol is a synchronized symbol. The synchronized symbol (Eyadeh 2008. 1) informs the receiver of the arrival of the packet which is used in calculating the time in which the data arrived. As the transmission progress, the transmitter controls the timing of each bit transmission in order for the receiver to maintain synchronization.

The transmission of data is dependent on the effectives of how appropriate synchronization is achieved. Correct synchronization happens through the process of organizing data into reliable synchronization frame. Each frame includes a synchronization symbol followed by a data symbol. (Eyadeh 2008. 1)

The receiver samples the value of the incoming signal at the center of each bit when synchronized. Caution must be taken to reduce skew in the sampling rate and timing. Individual bits that are extracted from the radio must be regrouped into blocks that are the encoded version of real data messages. Eventually, the blocks are decoded back into the original data and assembled into a packet. Communication systems should include error control systems to detect errors and to correct them.

2.3 Challenges in Wireless Sensor Networks

Recent improvements in the field of sensing, computing and communications have resulted in research efforts and high investments from various sectors in the area of WSNs. There are many issues that have influence in the design and performance of WSNs but only few have been presented

2.3.1 Hardware and operating system for WSN

WSNs are composed of hundreds of thousands of tiny devices called nodes. Sensor senses information, processes the information and passes the information to a Mote. A mote behooves of a processor, memory, battery, A/D converter for connecting to a sensor and a radio transmitter for forming an adhoc network. Fig. 2.2 shows the block diagram of a sensor node. (Gowrishankar, Basavaraju, Manjaiah & Subir 2008, 2.)

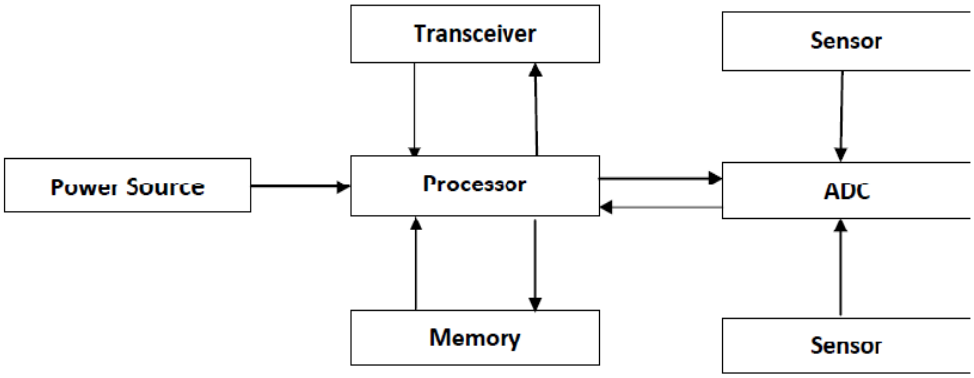


FIGURE 2.2 Structure of a senor node.

The nodes commonly found in sensor networks are small and limited in the consumption energy thereby placing a large constraints on how the hardware is designed

- Radio range of nodes is expected between 1 to 5 kilometers and paramount for making certain network connectivity and data collection.
- Energy consumption of the sensing device should be minimized and sensor nodes should be energy efficient since their limited energy resource determines their lifetime. (Gowrishankar, Basavaraju, Manjaiah & Subir 2008, 2.)

An operating system framework for a sensor node should be able to provide memory management and resource management in a restricted environment. OS design issues in WSNs are presented as follows:

- A sensor node is mainly responsible for computation of the extracted data from the local environment. It processes the extracted data and manipulates the data as required by the application.
- To minimize the consumption of battery energy certain components must be available. (Marco 2008, 17.)

2.3.2 Wireless Radio Communication Characteristics

The ability of sensor networks to communicate effectively and uniformly depends on the quality of the wireless communication systems available. Certain factors impede wireless communication such as diffraction, interference, vehicle penetration and building penetration. The basic block of communication in designing WSN should fulfill the following requirements:

- Low power consumption in sensor networks is imperative to enable batteries to last longer thereby fostering low duty cycle operation and local signal processing. (Gowrishankar, Basavaraju, Manjaiah & Subir 2008, 9.)
- Points to point communication in networks require high transmission power which causes data to be lost during transmission. In light of this, short range transmission should be considered to prevent loss of data.

- Systems used for communication should have components and features to control the subsystems to detect and correct errors and to correct as they are made. (Gowrishankar, Basavaraju, Manjaiah & Subir 2008, 9)

2.3.3 Localization

Localization in WSN is imperative for the management of a network and operation. In some instances, sensors are carried out without adequate knowledge of their relative position. In addition, there are no secondary infrastructures available to locate and handle them once deployed.

Finding the physical location of a sensor after they have been implemented is known as the problem of localization. The discovery of location of a sensor network should satisfy the following:

- The whereabouts' of node location can be used to implement energy efficient message routing protocols.
- Principles exhibited in localization must be exact, expandability and back mobility of nodes as they travel through the network.
- The algorithms must be able to localize failure, loss of nodes and tolerant to errors in physical measurements when calculating the distances between nodes. (Gowrishankar, Basavaraju, Manjaiah & Subir 2008, 9) (Bachrach & Taylor, 2)

2.3.4 Quality of Service

Quality of Service is the service provided by the sensor networks to its intended users. QoS for sensor networks is the optimum number of sensors sending information towards information-collecting sinks or a base station. Nodes have the capacity to join, leave and rejoin and links can be possibly broken thereby sustaining and reinstating the route dynamically is a constraint in WSNs. The main parameters in QoS are delay (latency), Jitter and offset. (Gowrishankar, Basavaraju, Manjaiah & Subir 2008, 8, 9.)

QoS issues in Wireless Sensor Networks are stipulated as follows

- Sensor networks should be supplied with the right amount of bandwidth to authorize it to accomplish the required QoS.
- QoS in WSN is complex to ascertain because the network topology may change continually and the available state information for routing inherently precise. (Gowrishankar, Basavaraju, Manjaiah & Subir 2008, 8, 9.)
- “Traffic unbalance in sensor network due to data aggregation from many nodes to a sink node. QoS mechanism must be designed for an unbalanced QoS constrained traffic”. (Gowrishankar, Basavaraju, Manjaiah & Subir 2008, 9.)

2.3.5 Security

Security is an important element to be considered critically when implementing Sensor Networks. Without security, vital information can be lost and stolen by a third party. Security is challenge in WSN since they are deployed in many areas such institutions and other establishments which require monitoring.

Sensor Networks should conform to these basic security demands such as:

- Confidentiality ensures vital data and information is protected and not exposed to unauthorized third parties. Confidentiality is needed to prevent information from being leaked when travelling from one to another. (Walters, Liang, Shi, Chaudhary, 2006. 6)
- “Lack of integrity may result in inaccurate information. Many sensor applications such as pollution and healthcare monitoring rely on the integrity of the information to function”. (Gowrishankar, Basavaraju, Manjaiah & Subir 2008, 9.)
- Since confidentiality and data integrity are essential, there is a need also to ensure the freshness of each message as they arrive. Data freshness means that the data is recent and no old messages have been replayed. It is crucial requirements when there is a shared-key strategy employed in the architect. (Walters, Liang, Shi, Chaudhary, 2006. 6)

2.4 Characteristics of WSNs

WSN have some distinctive characteristics and limitations that set it apart from all other networks and systems. Presented below are some of the distinctions:

- Sensor nodes are closely disposed and vary in association to the magnitude being higher than that of other networks.
- A sensor node does not depend on one another for configuration thus they configure themselves independently into a network.
- They are subtle to physical damages or failures due to its implementation in a hostile environment.
- Sensor nodes are dependent on battery making it difficult to change or recharge the batteries.
- Network topology changes from time to time because of certain factors such as node failures, damage, addition, energy depletion and channel fading. (Shio Kumar, Singh, D K Singh 2010, 2.)

2.5 Algorithm for Wireless Sensor Networks

Sensor applications require the communication of nodes to follow certain prescribe instruction to enable the clocks to be synchronized faster. Certain algorithms should be in place for nodes to know the exact task they are to perform. Examples of such algorithms are

- **Centralized Algorithms:** They are executed in a node that posses the knowledge of the whole network. These algorithms are quite rare because of the cost of transmitting the data to make the node know the status of the complete network.

- **Distributed Algorithms:** The communication is supported by message-passing. Since, WSN has limited amount of energy distributed algorithm protocols help in transmission delay in sending data among nodes

necessitating the increase of energy in the network. (Kiyani, Aghaee, Tahmasebi 2011, 1)

- **Location based Algorithms:** The nodes use restricted data acquired from a close area. Node localization assists in reporting sources of events, group querying of sensors, routing and locating network coverage. (Amitangshu 2010, 1)

3. TIME SYNCHRONIZATION

Time plays an important role in the operation of any distributed systems and in wireless sensor networks in particular, since they are supposed to observe and interact with physical phenomena. Synchronization in a sensor network aims to provide a common timescale for local clocks of nodes in the network. A global clock in a sensor system will help process and analyze the data correctly and predict future system behavior.

Some applications that require global clock synchronization are environment monitoring, navigation guidance, vehicle tracking etc. A clock synchronization service for a sensor network has to meet challenges that are substantially different from those in infrastructure based networks.

3.1 Relevance of Time Synchronization in WSN

Most of the clocks in servers and other related networking devices keep track of inaccurate time. The major reason for this deficiency is that designing a computer to store accurate time is not of importance to system manufactures because of cost and complexity. Moreover, even fairly accurate clocks are likely to vary due to manufacturing defects, changes in temperature, electric and magnetic interference. (Deeths 2001, 4.)

The smallest errors in keeping time add up over a long period of time. Most of the errors that take place in computer clocks cannot be quantified easily. Differences in clocks are caused by environmental, electronic variations and miscalibrated clocks (Deeths 2001, 4.). Time synchronization is insurmountable if clocks operate on the own their wavelength. Time synchronization is needed for accuracy and precision. Sensor nodes operating on a TDMA protocol need to agree on allocated time to prevent overlap and colliding of transmission which results in inaccuracies and error.

3.2 Time Synchronization Protocol

The majority of the applications of sensors networks depend on local clocks of sensor nodes for synchronization that rely on various degrees of precision. Properties of sensor

networks such as limited amount of energy, storage, computation and bandwidth in addition to potentially high density of nodes poses a challenge for WSN.

Due to these developments, in recent times, there have been researches carried out by institutions specifically focused on designing synchronization algorithms for sensor networks. This section presents in details the basic synchronization methods designed and proposed for sensor networks.

3.2.1 Reference Broadcasting Synchronization (RBS)

RBS proposed by Elson et al (Elson, Girod, Estrin 2002, 2) reduces the uncertainty of the sender by taking away the sender from the critical path. By removing the sender, the only ambiguity is the propagation and receives time. Majority of the synchronization protocols deploy sender to receiver synchronization which involves the sender sending timestamp information to the receiver for synchronization. (Ranganathan, Nygard 2010, 2.)

RBS has clear distinction because it uses receiver to receiver synchronization. RBS involves a third party broadcasting a beacon to all receivers (A and B) as shown in Fig. 3.1 The beacon contains no timing information, instead the receiver computes and compares their clocks (ta and tb) to find their relative phase offset. The timing is based on when a node receive the reference beacon. (Ranganathan, Nygard 2010, 2.)

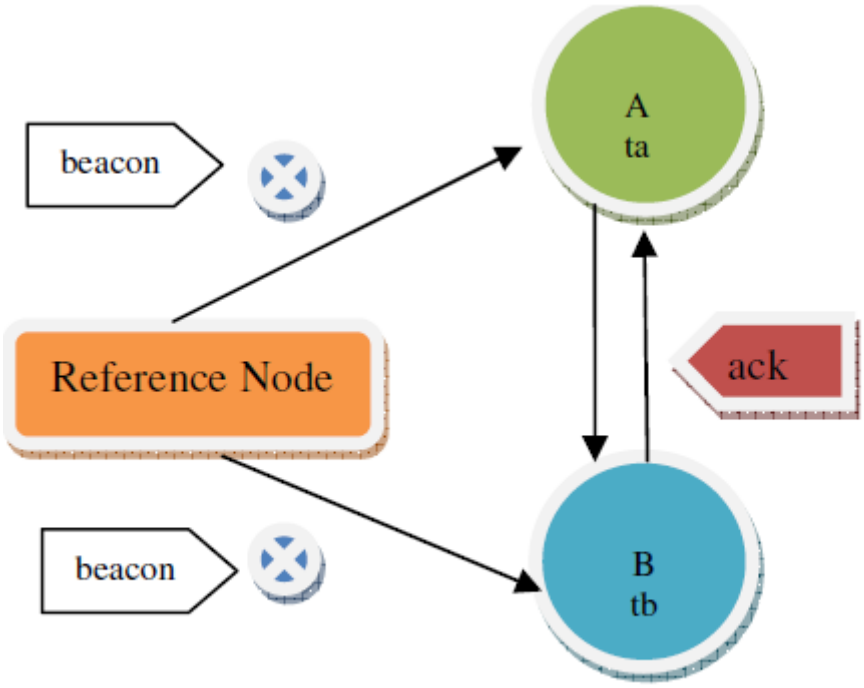


FIGURE 3.1 RBS Scheme. (Ranganathan, Nygard 2010, 2)

3.2.2 Timing Synchronization Protocol for Sensor Networks (TSPN)

TSPN is a sender and receiver based protocol which is needed to maintain accuracy when the number of node on a network increases. It operates on two phases namely Level discovery phase and Synchronization phase. At the discovery phase, the phase generates a hierarchical topology in the network in which each node is designated a level. A root node is given a level of 0. (Vikström 2006, 3. & Ranganathan, Nygard 2010, 3).

A node is designated level *i* to a node of level *i*-1 in the Synchronization phase. When the synchronization phase draws to a close, all the nodes are synchronized to the root node and the network-wide synchronization is accomplished. (Vikström 2006, 4 & Ranganathan, Nygard 2010, 3).

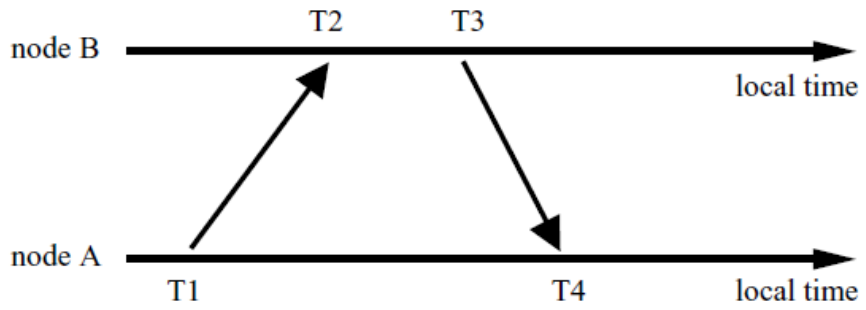


FIGURE 3.2 Synchronization phase (Sivrikaya, Yener 2004, 4)

Fig 3.2 shows the synchronization phase in which node A starts the synchronization process by transferring a synchronization pulse packet at T1 (that is, according to its local clock). The packet sent have A's level number and the current value of T1.

The sent packet arrives in B according to its local clock at T2 = T1 + Δ + ***d,*** where Δ is the relative clock drift between the nodes and ***d*** is the propagation delay of the pulse. When B receives the packet at time T3, it responds with an acknowledgement which includes the level number of B and the values of T1, T2 and T3. (Ranganathan, Nygard 2010, 4. & Sivrikaya, Yener 2004, 4) Node A computes the clock drift and the propagation delay as

Clock drift:

$$\Delta = \frac{(T2 - T1) - (T4 - T3)}{2} \tag{3.1}$$

Propagation delay:

$$d = \frac{(T2 - T1) + (T4 - T3)}{2} \tag{3.2}$$

3.2.3 Lightweight Time Synchronization Protocol (LTS)

The Lightweight Time Synchronization (LTS) protocol attempts to synchronize the clocks of a sensor network to the clocks held by certain reference nodes which may

have GPS receivers. LTS has no limitation with respect to the local clock update discipline and it does not try to estimate actual drift rates. (Sivrikaya, Yener 2004, 5)

LTS produces a low-depth spanning tree comprised of the nodes in the network and utilizes a pair-wise synchronization algorithm for each one of the two nodes on the same edge of the tree. (Sivrikaya, Yener 2004, 5)

LTS is divided into two building blocks namely:

- “A pair-wise synchronization protocol synchronizes two neighboring nodes”. (Karl, Willig 2003, 223 & Sivrikaya, Yener 2004, 9)
- To preserve the entire nodes synchronized to a common reference, a spanning tree from the reference node to all the nodes is created. (Karl, Willig 2003, 223 & Sivrikaya, Yener 2004, 9)

4. NETWORK TIME PROTOCOL

NTP is a system for distributing Coordinated Universal Time (UTC) by a way of synchronizing the clocks of computer systems over a packet switched variable-latency data networks.

The clients and servers used in the NTP have a capabilities and possibilities over a wide range of network delays and jitter characteristics. Most users of the internet NTP synchronization subnet use a software package including the full suite of NTP options and algorithms, which are relatively complex and real time applications.

NTP works on assumption of having all the system clocks get as close as possible to the correct time. NTP corrects the current time and makes sure the time is consistent on the devices and automatically adjusts for time drift on the client. When this is achieved, it allows for small network traffic and stores client clocks more reliably, even if the network is unreachable due to changes in topology or any mishaps. (Deeths 2001, 2)

NTP operates on different platforms. Simple Network Time Protocol (SNTP) is a lightweight variation of NTP which is compatible with NTP and NTP can be standardized the throughout an enterprise network because many platforms and networking devices support either NTP or SNTP. (Deeths 2001, 2)

4.1 Components of NTP

There are numerous protocols to synchronize computer clocks, each of which has distinctive features. Below are lists of NTP features:

- The NTP needs a reference clocks such as GPS, DCF77 and WWVB that define the true time to operate. All the clocks are set towards the true time. (It will not just make all systems agree on some time, but will make them agree upon the true time as defined by some standard).
- NTP utilizes UTC as reference time which is an official standard for current time. UTC evolved from the former GMT (Greenwich Mean Time) that once was used to set the clocks on ships before they left for a long journey.

- NTP is a fault tolerant protocol that possibly selects the best candidate of time sources to synchronize to.
- NTP is highly scalable and expandable. Individual node on the network trade time information either bidirectional or unidirectional. (The NTP FAQ and HOWTO 2006)

4.2 Operation of NTP

NTP employs a client-server architecture in which a time server distributes reference time to clients, which may in turn distribute the time to other hosts. The protocol itself is simple and does not specify how precisely servers and clients must synchronize or maintain time.

NTP uses the User Datagram Protocol (UDP) on port 123 to communicate between clients and servers. Several attempts are made at designate intervals until the server responds with a time. The time interval depends on certain factors and ranges. UDP helps to avoid retries from using up network bandwidth if a time server with a large number of clients is no longer available. (Deeths 2001, 7.)

4.2.1 Hierarchical Model

NTP operates on a hierarchical model in which a small number of servers give time to a large number of clients. The clients on each level are in turn a potential server to large number of clients of a higher numbered stratum.

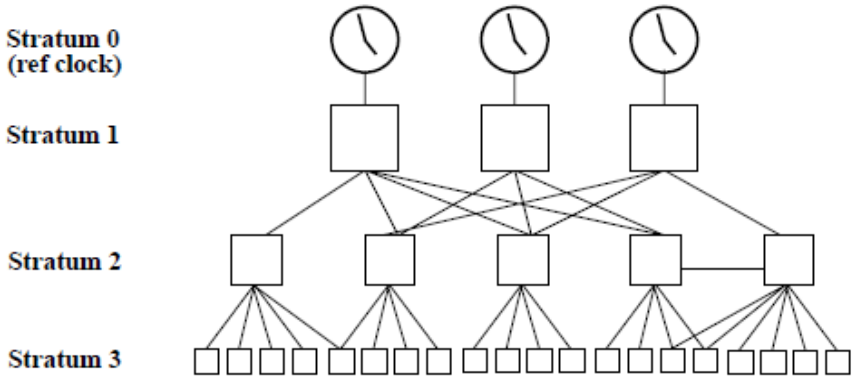


FIGURE 4.1 Hierarchical model of NTP. (Deeths 2001, 7)

Servers connected directly to a reference clock are termed stratum 1 servers. A reference clock having a relationship with a stratum 1 server is a stratum 0 server. Reference clock provide current time. Clients cannot have an effective communication with a stratum 0 server unless they are connected to a stratum 1 server which is synchronized to a stratum 0 server. (Deeths 2001, 8.)

Clients of stratum 1 servers are referred to as stratum 2 clients. If they serve time to clients, they are called stratum 2 servers, and the clients they server are stratum 3 clients. NTP can contain a stratum number of client up to 15. Reference clocks are situated on any NTP hierarchy. These are electronic clocks which are synchronized to a common time reference. Reference clocks are to be accurate and precise. The accuracy of other clocks is judged according to how close a clock is to a reference clock, the network latency to the clock, and the claimed accuracy of the clock. (Deeths 2001, 6, 8)

NTP performs statistical values that compared the time to choose the one that is close to proximity. These values are offset, jitter, frequency error and stability. Thereby, each NTP server maintains an approximation of the quality of its reference clock and itself. (The NTP FAQ and HOWTO 2006).

5. TOOLS

5.1 Meinberg NTP Software

Meinberg NTP Software is a software package that contains a background utility (daemon or services) which aids in synchronizing the computer system time to one or more external reference time sources which can either be devices on the network or a radio clock which is connected to a computer.

It provides accuracies within a millisecond on LANs and up to a few tens of milliseconds on WANs relative to Coordinated Universal Time (UTC) via a Global Positioning Service (GPS) receiver. Typical NTP configurations utilize multiple redundant servers and diverse network paths in order to achieve high accuracy and reliability. (Meinberg NTP Software 2008).

5.2 NTP Time Server Monitor

The NTP Time Server Monitor allows the configuring and controlling the local NTP service with a user friendly graphical user interface which displays the current status of the local NTP service and external NTP services. Fig. 5.1 shows the service of the NTP Time Server Monitor when it is started.

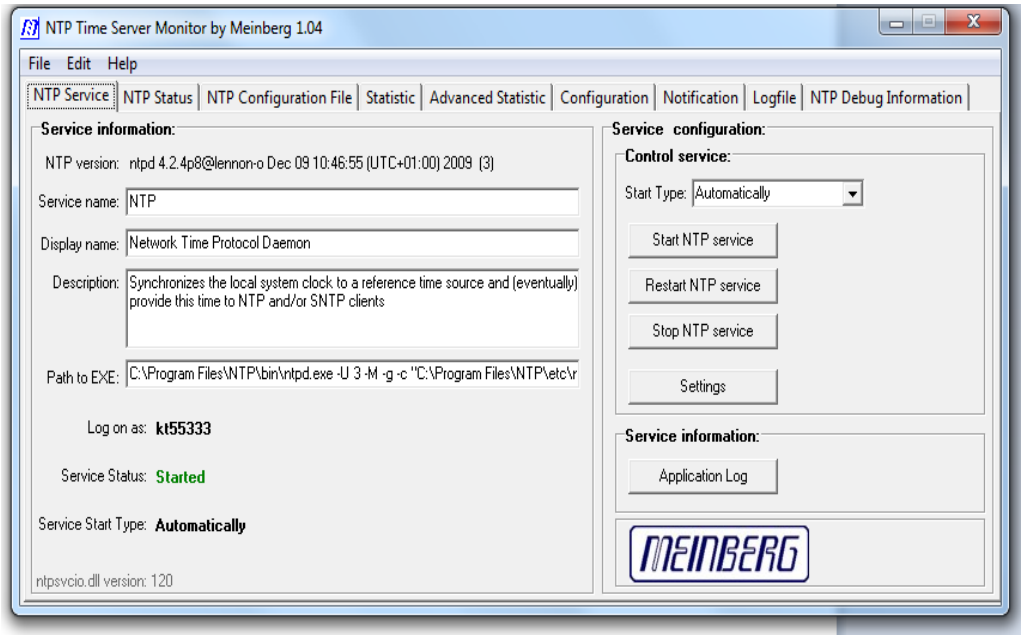


FIGURE 5.1 Service page

The software is able to control and supervise the NTP service of windows over a manageable graphical user interface. It reads the system parameters and displays them in a textbox for evaluation. Additionally the current offset, stratum value and the selected synchronization source is displayed in a status string. The status refreshes automatically every 10s. (Meinberg NTP Software 2008)

5.3 Roving WLAN Module (Rn-131g-eval)

The work was carried out with the help of Meinberg NTP Software and Roving WLAN module (Rn-131g-eval). The Rn-131g-eval module is a standalone, embedded wireless 802.11 networking module which has a lot of networking capabilities and functionalities. Rn-131g-eval comes in a small form factor and consumes low power thereby enabling it to perform on many mobile wireless applications such as

- Asset monitoring
- GPS monitoring
- Wireless audio
- Industrial sensors and controls
- Home automation and security

- Home automation and security
- Medical devices.

The WiFly GSX module combines 2.4 GHz radio, processor, TCP/IP stack, real-time, crypto accelerator, power management and analog sensor interfaces. It embodies networking applications such as DHCP, DNS, TCP, ICMP and sockets

When configuring, the hardware requires four connections (PWR, TX, RX and GND) to generate a wireless data connection. Fig 5.2, shows the block diagram of WiFly GSX. The sensor interface gives temperature, audio, motion, acceleration and other analog data without additional infrastructure. The WiFly GSX module is programmed and administered with a simple ASCII command language. Once the WiFly is setup, it can scan to locate an access point, associate, authenticate and connect over any WiFly network (Roving Networks 2009, 1).

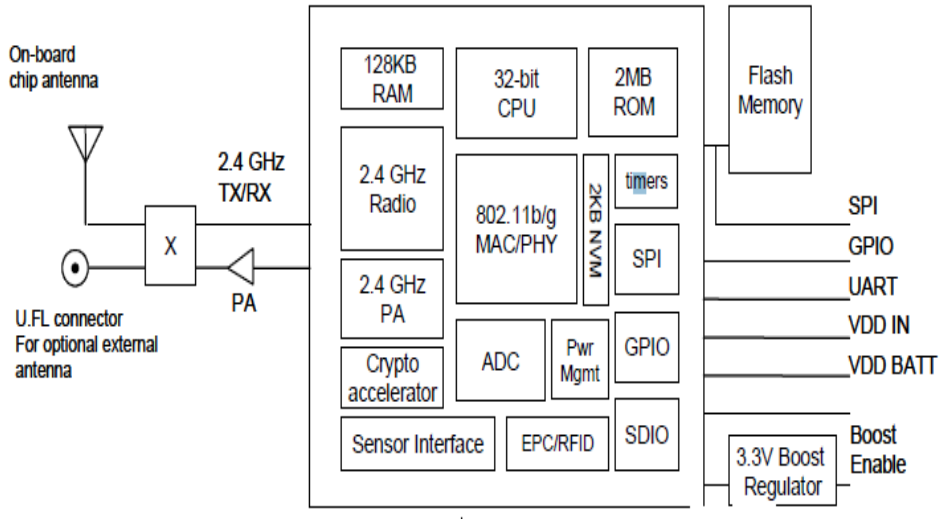


FIGURE 5.2 Block Diagram. (Roving Network 2009,1)

6. EXPERIMENT

6.1 Hardware Setup

6.1.1 Graphical Representation

The graphical representation in Fig 6.1 shows the equipment and tools that were used. It summaries the whole work and how the various devices were connected.

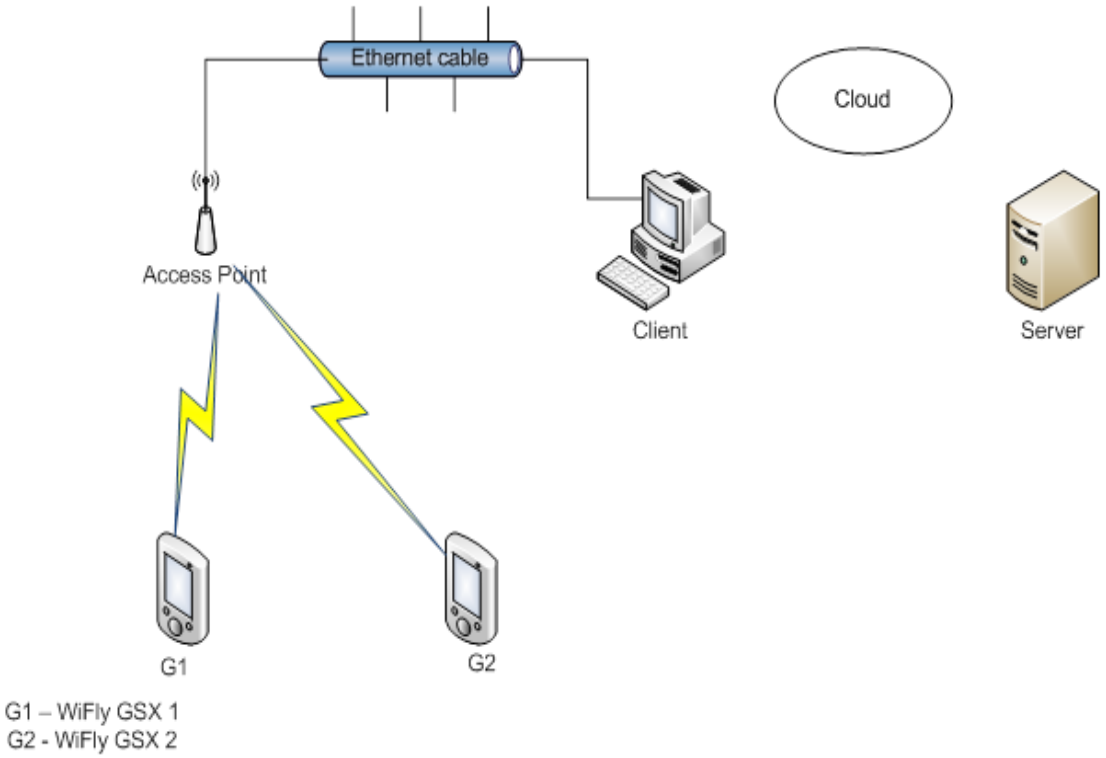


FIGURE 6.1 Graphical representation

6.1.2 Evaluation Board Setup

For the lab work to be implemented successfully, certain environmental conditions and electrical characteristics should be in place. Table 1 and Table 2 details the necessary conditions that were used

Table 1. Environmental conditions

Parameter	RN-131G
Temperature Range (Operating)	-30℃ ~ + 85℃
Temperature Range (Storage)	-40℃ ~ + 85℃
Relative Humidity (Operating)	≤ 90%
Relative Humidity (Storage)	≤ 90%

Table 2. Environmental characteristics

Supply Voltage	Min	Typ.	Maxi	Unit
Supply Voltage VDD	3.0	3.3	3.7	VDC
Supply Voltage (VBATT Option)	2.0	3.0	3.3	VDC
PIN 21 Switched 3.3V Output			150	ma
Digital Output				
Input logic HIGH VIH	2.3			VDC
Input logic LOW VIL			1.0	VDC
Digital Output Drive				
PIO 4,5,6,7,8		24		ma
PIO 9,10,11,12,13		8		ma
Power Consumption				
Sleep		4		μA
Standby (doze)		15		mA
Connected (Idle, RX)		40		mA
Connected (TX)		140		mA

The degradation of radio signal is influenced by many factors such as diffraction, refraction, vehicle and building penetration. These factors have more impact on the propagation of wireless signals with low power radios which are commonly found in WSNs. Table 3 stipulates the radio characteristics

Table 3. Radio Characteristics

Parameters	Specifications
Frequency	2402 ~ 2480 MHz
Modulation	802.11b compatibility: DSSS,DBPSK, OFDM
Channel Intervals	5MHZ
Channel	1 – 14
Transmission rater (Over Air)	1 – 11Mbps for 802.11b / 6 – 54Mbps for 802.11g
Receive sensitivity	-85 dBm typ
Maximum RF Input to UFL connector	10 dm

The evaluation board in Fig 6.1 was connected to the computer through the WiFly GSX module. The drivers were installed which enabled the programmer to appear as a USB Serial Port on a COM port. The Evaluation board started with a green LED when inserted into the computer USB port as shown in Fig 6.2. The power that was used was provided to by soldering the 9-battery clip wire to J7 on the board and a 9V battery was attached to the clip. 10-pin serial cable was connected to J3 on the board. The battery triggered a blue and green LED to blink slowly. The LED on the Evaluation board has some indicators which ignite when the module is correctly configured. Table 4 shows the various LED indicators on the board.

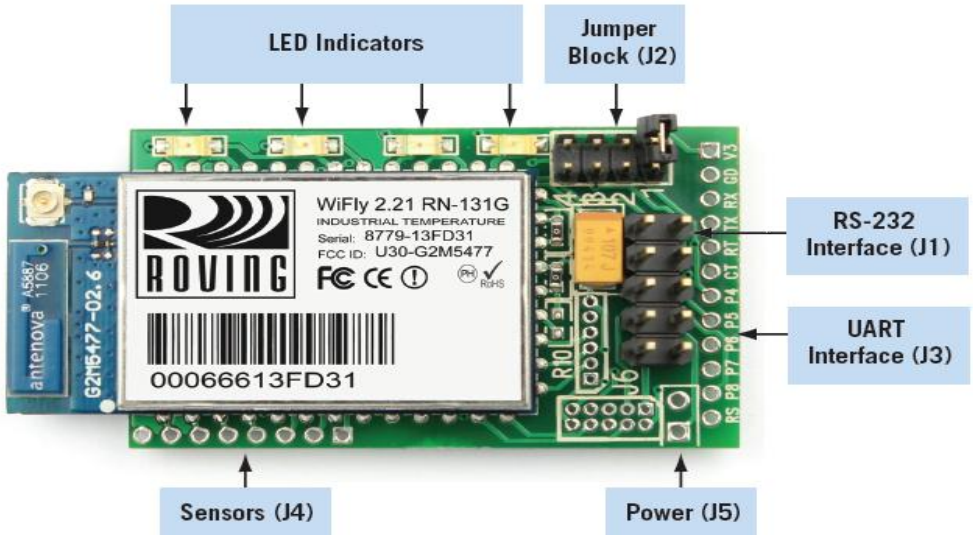


FIGURE 6.1 Evaluation board (Roving Networks 2009, 5)

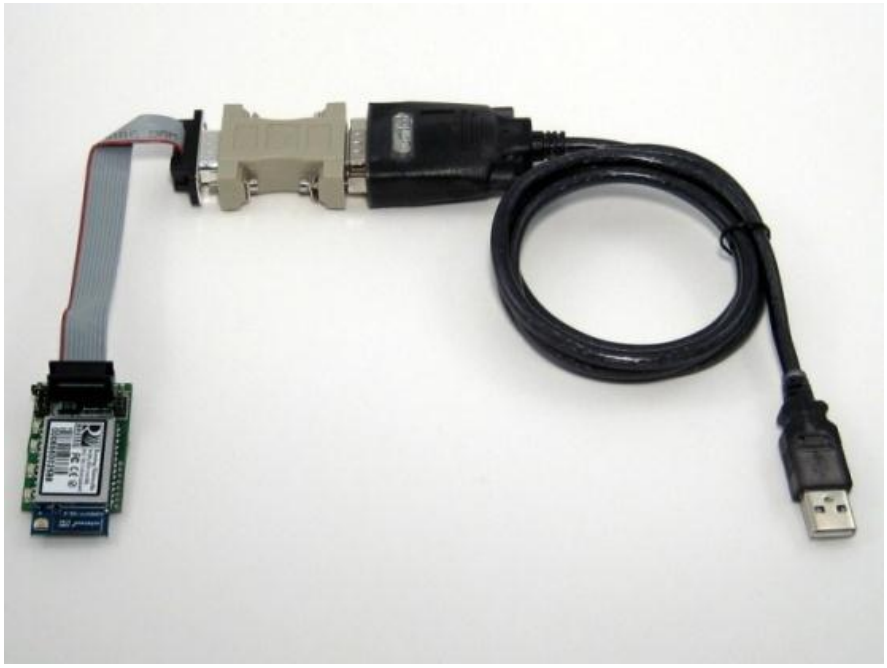


FIGURE 6.2 Evaluation board connected to USB

Table 4, LED Indicators

Conditions	Blue LED	Red LED	Yellow LED	Green LED
On Solid	Not Used	-		Connected over TCP
Fast blink	Not Used	Not Associated	Rx/Tx,data Transfer	No IP address
Slow blink	Not Used	Associated, no internet		IP address ok
Off	Not Used	Associated, internet ok		

6.2 Configuring Rn-131g-eval

The Rn-131g-eval modules function in two distinctive modes: Data mode (default) and Command mode as shown in Fig 6.3. When the Rn-131g-eval is operating in the data mode, the WiFly module behaves as a data pipe. The module accepts data, processes

it over a Wi-Fi, and encapsulates the TCP/IP header and trailers. It later transfers the user data to the Universal Asynchronous Receiver/Transmitter (UART). The module works on the TCP/IP packet by processing the stored data and transfers it out over a Wi-Fi. The end processor is aware of the process of sending and receiving data to the host. (Roving Networks 2011, 6)

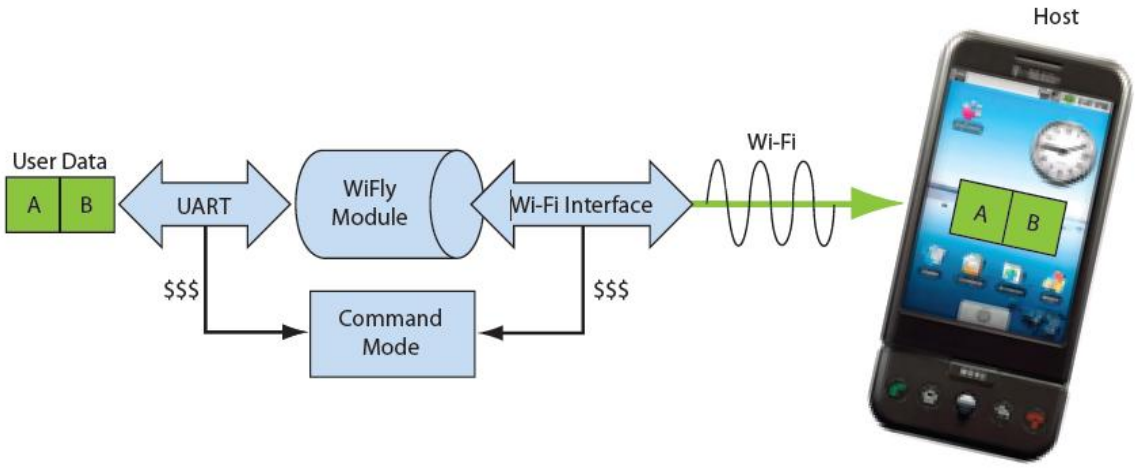


FIGURE 6.3 Data mode and Command mode (Roving Networks 2011, 6)

The WiFly module is in a data mode by default thereby typing the escape sequences $$$ enabled the module to enter the command mode for configuration to take place. When it entered the command mode, the configuration was deployed with ASCII Command. Misappropriation of the ASCII command resulted in errors which resulted in debugging and troubleshooting to ascertain the source of the error.

6.2.1 Configuring Module Through UART

In order for communication to emerge with the module via a Terminal Emulator, the COM port was allocated to the USB of the module. The terminal emulator utility program was opened and the COM port was specified. Since Tera Term was used during the work, the Serial and chose COM port 2 and 22 were respectively selected from the port drop down listbox as shown in Fig. 6.4

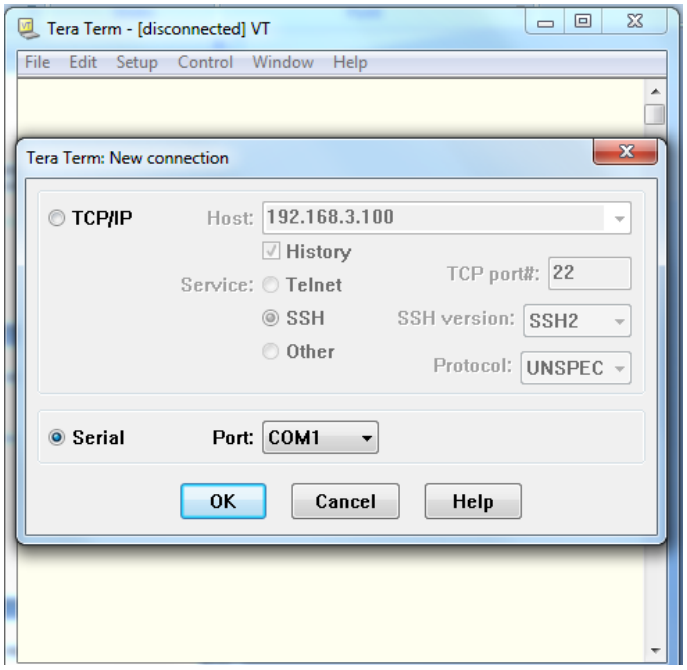


FIGURE 6.4 COM port assigned

6.2.2 Associating with an Access Point

The Tera Terminal window was opened, but it was in the data mode. The escape sequence $$$ put the module in the command mode. By typing the ***show net***, the current network settings on the network were displayed

```
<2.21> show net
SSid=myfinalwork
Chan=6
Assoc=OK
Rate=12, 24Mb
Auth=OK
Mode=OPEN
DHCP=OK
Boot=9304735
Time=OK
Links=4
```

When the module is in the command mode, it has the capability of searching and finding all available networks within its circumference. The ***scan*** command was used to perform the task of finding available networks. The network ***myfinalwork*** is an open access point which is a suitable candidate for forming an access point.

```
<2.21>
SCAN:Found 7
Num           SSID   Ch  RSSI   Sec    MAC Address     Suites
 1            Savonia 01 -47 WPA2PSK 00:1a:2f:91:3e:a0 AESM-AES  3104    0
 2          Savonia-web 01 -48 WPA2PSK 00:1a:2f:91:3e:a1 AESM-AES  3104    0
 3            LangeDemo 01 -66    Open c0:c1:c0:3f:cc:05    100    4
 4         GainSpanDemo 06 -44    Open 98:fc:11:47:40:c7    104    4
 5          myfinalwork 06 -21    Open 00:23:69:c4:de:43   2100    0
 6          Savonia-web 11 -69 WPA2PSK 00:1a:2f:91:48:21 AESM-AES  3104    2
 7          Savonia-web 11 -59 WPA2PSK 40:f4:ec:12:0f:61 AESM-AES  3104    0
```

In order to associate with an access point, ***join myfinalwork*** was typed. The red LED goes off signifying the module was successfully configured to an access point and the green LED flashes slowly.

```
<2.21> join myfinalwork
DeAuth
Auto-Assoc myfinalwork chan=6 mode=OPEN SCAN OK
Joining myfinalwork now..
<2.21> Associated!
Using Static IP
```

Since myfinalwork was successful associated to a network, the access point SSID was stored for future purposes. This enables the pass phrase to be solved to the config file for the module to associate with the network whenever it is booted up.

The pass phrase for Wi-Fi Protected Access (WPA) which provides authentication and encryption was configured as

set wlan channel 6

set wlan join 1

Set wlan phrase wsnfinalwork

Save

Reboot

6.2.3 Configuring Ad-Hoc Mode

Adhoc is a point to point network in which each individual Wi-Fi device is linked directly to every other Wi-Fi device on the network. There is no availability of access point. In adhoc network, the Wi-Fi devices that are available cooperate in preserving the network

from malfunctioning. The WiFly acted as an access point enabling other Wi-Fi devices to be associated hence enhancing communication processes.
Configuring Adhoc mode can be done in two different ways either by Hardware or software. During the course of the work, the Adhoc mode was necessitated through software.

From the command mode, the module was configured for adhoc mode with the ***join*** command.

set wlan join 1

set wlan ssid myfinalwork

set wlan chan 6

DHCP was turned OFF to disable automatic IP assignment. IP address and the netmask were manually assigned so that other devices know where to connect to the adhoc WiFly GSX.

set ip address 192.168.3.101

set ip netmask 255.255.255.0

set ip dhcp 1

After the IP address and netmask were configured, the configurations were saved and reboot ushering the module into the adhoc mode. Immediately, the red LED began to blink fast indicating it is associated with an access point whereas the yellow LED automatically goes OFF while the green LED continuously blinks signifying the IP address is set.

6.2.4 Configuring Automatic Connections

In order not to be assigning IP address and netmask anytime the system is powered on, the module was configured automatically, so that it can connect to a remote server which is any computer providing information/data to other computers, send data and later disconnect from the host. This was achieved with these parameters

set ip host 192.168.3.189

set ip remote 55555

set sys autoconn 1

set com idle 5

set sys sleep 2

set sys wake 60

save

reboot

6.2.5 Real Clock Configuration

When the module is powered on, there is a real clock function which has the capability of knowing the number of seconds and the actual time when synchronized with SNTP time server. The module keeps track of up time but it does not synchronize with the time server which requires being associated with a network that can access the SNTP server. I made sure the module was associated with a network (myfinalwork) for connections to be made with the SNTP time server. **E*nable 1*** command was configured to help the system acquire time from the server when powered on at anytime.

```
<2.21> set time enable 1
AOK
<2.21> get time
TimeEna=1
TIMEADR=192.168.3.189:123
Zone=3
```

By issuing the command ***show t t***, all the necessary time variables were displayed

```
<2.21> show t t
Time=07:59:39
UpTime=294 s
RTC=7821682075419 ms
Restarts=1
Wake=1
```

6.3 Configuring NTP

NTP client performs their duties with the server through mutual understanding and engaging in communication. The client initiates the process by sending a message in which the client responds with the sent message and an acknowledgement. The NTP configuration file located at ***etc/inet/ntp.config*** houses configuration data for the NTP daemon. This section divulges into in configuring clients, server, access control and authentication.

6.3.1 Configuring NTP Client

NTP client on the hierarchical model consists of a number of servers, broadcast and non-broadcast servers that has the propensity to be used by the same client. The main responsibility of the NTP client is to synchronize their allocated time in correspondence to the NTP server. The NTP servers do not synchronize their time in consonance to the NTP client because NTP clients can also perform the duties of NTP servers to clients of their own accord (Deeths 2001, 4).
Configuring NTP client behooves of the following

server ns1.savonia.fi

server ntp.a-lab.fi

To avoid misunderstanding, NTP clients are configured with the server keyword. Two or more servers are used so that the one which the preferred time is chosen for synchronization. When the preferred server fails which can be due to many factors, the secondary server takes over the synchronization process.

6.3.2 Configuring NTP Server

Configuring NTP server involves setting up a peer such as

peer 192.168.3.189

peer 192.168.3.100

Drift file contains important information on the rate of the drift of a client clock (Deeths 2001, 4). When operational, the NTP daemon (***xntpd***) acquires knowledge about the drift of the system clock in parallel to the reference clock. (The NTP FAQ and HOWTO 2006).

For ***xntpd*** to commemorate the drift which is updated every hour, this command was added to the configuration file

driftfile /etc/ntp.drift

driftfile "C:\Program Files\NTP\etc\ntp.drift

One of the merits of using a drift – file is that, xntpd always make good use of the last written value as initial frequency correction after the system is restart. In this situation, the appropriate correction is configured faster because in absence of a drift-file, the initial frequency correction amounts to zero. (The NTP FAQ and HOWTO 2006).

Broadcast and multicast servers were configured and were delineated with *broadcast* command in the ntp.config file. The broadcast address which sends data/information to all devices connected to a network was configured and specified after the broadcast command.

broadcast 192.168.3.255 ttl 6

broadcast 224.0.0.1 (multicast address)

6.3.3 Configuring Authentication

For authentication to emerge and progress in NTP, both the client and the server must have a keys.conf file stipulated in etc/inet/ntp.config involving the key with the same key ID. The client and server were configured for authentication because the client/server configurations were carried backwards to enable broadcast and non-broadcast.
Configuring authentication both the server and the client require the following steps

keys / etc/ inet/ ntp.keys

server ns1.savonia.fi prefer key 1

server ntp.a-lab.fi key 1

trustedkey 1

enable auth

broadcast 192.168.3.255 key 12 ttl 6 (broadcast configuration)

broadcast 224.0.0.1 key 14 ttl 6 (multicast configuration)

The ***trustedkey*** keyword was for configuring the broadcast and multicast servers so that the servers will be able to acknowledge clients during initiation processes. This takes place in order for the clients to ascertain how fast the network is operating.

The ***peers*** were also configured for authentication since they are all likely candidates for initiation. The necessary keys used were defined as trusted to aid in communication and synchronization.

keys / etc/ inet/ ntp.keys

trustedkeys 2 1

peer 192.168.3.189 key 2

peer 192.168.3.100 key 1

6.3.4 Managing Effect of Network

NTP architects should be able to comprehend the working mechanisms of NTP and the underlying networks that are associated. Minimizing NTP's network operation and maximizes the accuracy and precision of the clocks are the prime objective of the NTP architects. For high clock accuracy and precision to be obtained, the network latency should be calibrated to a low degree. NTP was able to accomplish a high level of accuracy and a good network candidate since local NTP servers. One of the ways accuracy was achieved was to minimize the latency between the various connections by inserting NTP servers on the same LAN as their clients. (Deeths 2001, 16)

6.3.5 Configuring Access Control

Since the world is now populated with hackers trying to harness vital information/data belonging to institutions and individuals, it is imperative for monitoring to be turned off because it gives them the opportunity to gather sensitive information about hosts or networks. Restricting access to NTP queries blocks intruders and hackers the privilege of exploring information with ***ntpd*** command. Sensitive information to be retrieved by an intruder involves network delays which determine the network architecture, hostnames, IP addresses and Operating System version. (Deeths 2001, 17)
The function ***restrict*** keyword defined in the ntp.config, is used by NTP to restrict access to its services and functions.

 # allow localhost
restrict 127.0.0.1
restrict 192.168.3.189

#accept packets from

restrict 192.168.3.100 mask 255.255.255.0 nomodify notrap
restrict 192.168.3.101 mask 255.255.255.0 nomodify notrap
restrict 192.168.3.189 mask 255.255.255.255 nomodify notrap noquery

restrict 192.168.3.100 mask 255.255.255.255 nomodify notrap noquery

Access control mechanism and authentication provide a strong security establishment for NTP hence preventing intruders and hackers from gaining sensitive information.

7. RESULTS

Wireshark, a network packet analyzer, was installed on the host computer to assist in capturing network packets and displaying the packet data with its technical information. When the host was connected to a network through a fast Ethernet, the Wireshark captured and displayed the result shown in Fig. 7.1

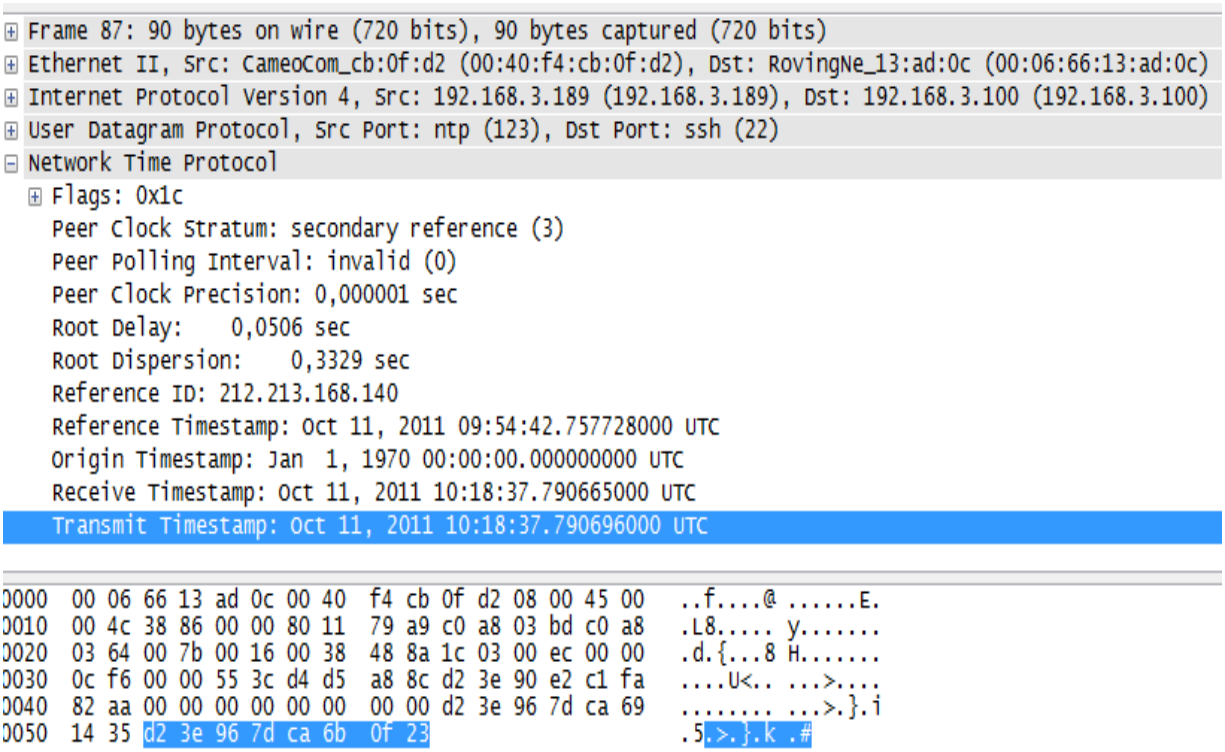

Frame 87: 90 bytes on wire (720 bits), 90 bytes captured (720 bits)
Ethernet II, Src: CameoCom_cb:0f:d2 (00:40:f4:cb:0f:d2), Dst: RovingNe_13:ad:0c (00:06:66:13:ad:0c)
Internet Protocol Version 4, Src: 192.168.3.189 (192.168.3.189), Dst: 192.168.3.100 (192.168.3.100)
User Datagram Protocol, Src Port: ntp (123), Dst Port: ssh (22)
Network Time Protocol
Flags: 0x1c
Peer Clock Stratum: secondary reference (3)
Peer Polling Interval: invalid (0)
Peer Clock Precision: 0,000001 sec
Root Delay: 0,0506 sec
Root Dispersion: 0,3329 sec
Reference ID: 212.213.168.140
Reference Timestamp: Oct 11, 2011 09:54:42.757728000 UTC
Origin Timestamp: Jan 1, 1970 00:00:00.000000000 UTC
Receive Timestamp: Oct 11, 2011 10:18:37.790665000 UTC
Transmit Timestamp: Oct 11, 2011 10:18:37.790696000 UTC
0000 00 06 66 13 ad 0c 00 40 f4 cb 0f d2 08 00 45 00 ..f....@ ......E.
0010 00 4c 38 86 00 00 80 11 79 a9 c0 a8 03 bd c0 a8 .L8..... y.......
0020 03 64 00 7b 00 16 00 38 48 8a 1c 03 00 ec 00 00 .d.{...8 H.......
0030 0c f6 00 00 55 3c d4 d5 a8 8c d2 3e 90 e2 c1 fa ....U<.. ...>....
0040 82 aa 00 00 00 00 00 00 00 00 d2 3e 96 7d ca 69 ....>.}.i
0050 14 35 d2 3e 96 7d ca 6b 0f 23 .5.>.}.k .#

FIGURE 7.1 Captured packets displayed by Wireshark.

From the Meinberg NTP Time Server Monitor, the current status of NTP service was displayed with ***ntpd –p*** as shown in Fig. 7.2

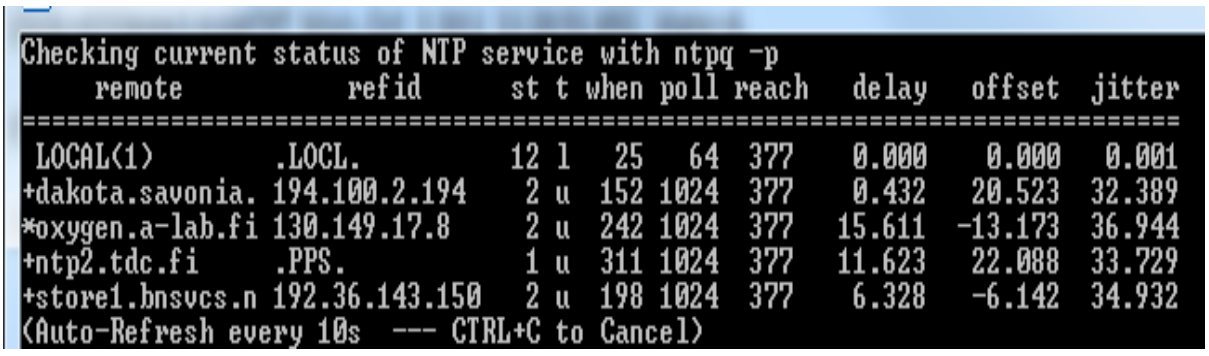

Checking current status of NTP service with ntpq -p
remote refid st t when poll reach delay offset jitter
LOCAL(1) .LOCL. 12 l 25 64 377 0.000 0.000 0.001
+dakota.savonia. 194.100.2.194 2 u 152 1024 377 0.432 20.523 32.389
*oxygen.a-lab.fi 130.149.17.8 2 u 242 1024 377 15.611 -13.173 36.944
+ntp2.tdc.fi .PPS. 1 u 311 1024 377 11.623 22.088 33.729
+store1.bnsvcs.n 192.36.143.150 2 u 198 1024 377 6.328 -6.142 34.932
(Auto-Refresh every 10s --- CTRL+C to Cancel)

FIGURE 7.2 Current status of NTP

+Dakota.savonia.fi, +ntp2.a-lab.fi and + store1.bnsvcs.n are secondary candidates for synchronization. NTP will compare the three and the choose the one with a high degree of accuracy when the oxygen.a-lab.fi fails or when there is a change in network topology.

***oxygen.a-lab.fi** was the candidate for synchronization since it provides better accuracy and precision. The reference clock was synchronized to it and time was distributed to the clients as shown in Fig 7.3 and Fig 7.4

Time on COM port 2

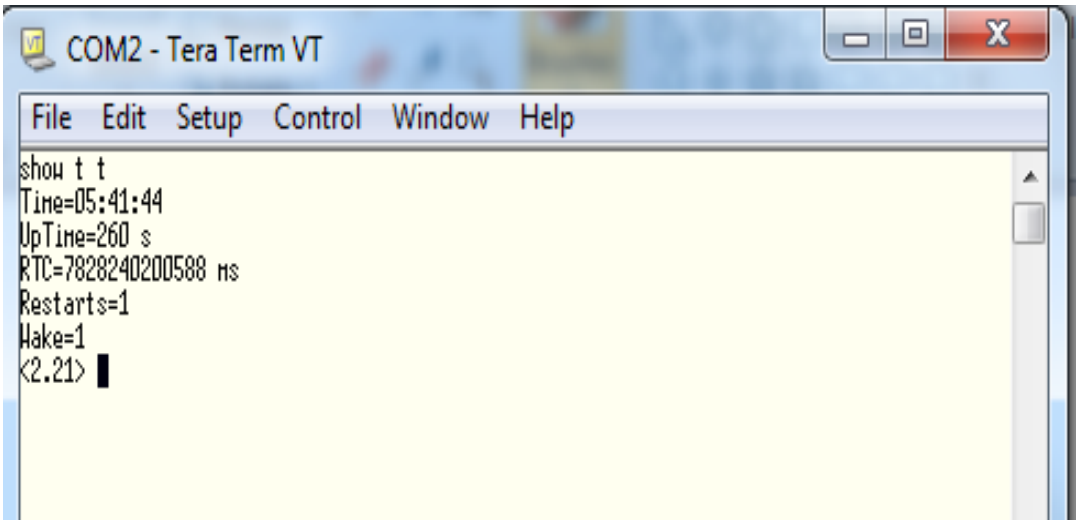


FIGURE 7.3 Time on port 2

Time on COM port 22

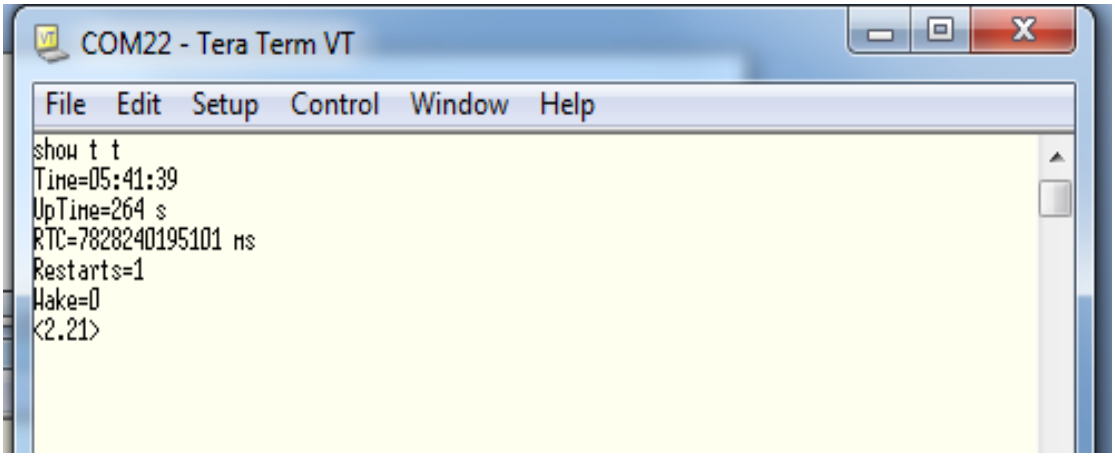


FIGURE 7.4 Time on port 22

8. CONCLUSION

In this work, the transmission and processing of information in sensor networks were explained; the various Time Synchronization protocols such as RBS, LTS, TPSN and NTP were fully examined. The WiFly modules (G1 and G2) were provided with time 05:41:41 and 05:41:39 respectively from a device (Client) that was synchronized to an external reference clock (Server). From careful observation, the provided time was not accurate since there was a delay of 2 milliseconds which was due to propagation, transmission and queuing delay.

The importance of keeping time accurate and precise on multiple devices is a critical challenge in a distributed network. A slight deviation in time can cause much havoc in organizations since time is crucial for recording and updating events as they happen. In light of this development, all the devices on the network need to be synchronized to an external reference source known for its accuracy for time to be distributed wirelessly to all other devices directly connected.

Financial Institutions and cooperate organizations that rely heavily on the accuracy of time for daily operations should deploy NTP on their networks. Sensor Networks are in their early stages of development necessitating for more research and development to design low cost, energy efficient and high performing sensors. Since NTP has some drawbacks, modifying and improving the current synchronization protocols such as NTP in today’s networked environments can help to improve efficiency.

REFERENCES

Chao-Nong, Xu. Zhao, Lei. Yong-Jun, Xu & Xiao-Wei, Li. 2006. *Simsync: A Time Synchronization Simulator for Sensor Networks* [online pdf]. [accessed 09 November 2011]. Available from: http://www.aas.net.cn/qikan/manage/wenzhang/060620.pdf.

Deeths, D. 2001. Enterprise Engineering and Glenn Brunette – SunPS. *Using NTP to control and Synchronize Systems Clocks* [online pdf]. Part I: Introduction to NTP [accessed 02 February 2012]. Available from: http://anselmo.homeunix.net/Sun-BluePrints/0701/NTP.pdf

Deeths, D. 2001. Enterprise Engineering and Glenn Brunette – SunPS. *Using NTP to control and Synchronize Systems Clocks [online pdf].* Part II: Basic NTP Administration and Architecture. [accessed 02 February 2012].
Available from: http://lx.ujf.cas.cz/misc/time/sun-blueprint/ntp2.pdf

Elson, J. Girod, L. & Estrin, D. 2002. *In Proceedings of the Fifth Symposium Operating Systems Design and Implementation (OSDI 2002)* [online pdf]. Fine-Grained Network Time Synchronization using Reference Broadcasts. [accessed 20 December 2011]. Available from: http://lecs.cs.ucla.edu/Publications/papers/broadcast-osdi.pdf

Eyadeh, A. A. 2008. *International Journal of Communications, Issue 1. Volume 2.* [online journal]. Frame Synchronization for an OFDM System. [accessed 27 December 2011]. Available from: http://www.naun.org/journals/communications/c-43.pdf

Gautam, G. C. & Sharma, P.T. 2010. *International Journal of Applied Engineering Research, Dindigul. Vol. 1, No.4, 2011.* [online pdf]. A Comparative Study of Time Synchronization Protocols in Wireless Sensor Networks. [accessed 15 November 2011]. Available from: http://research.ijcaonline.org/volume36/number11/pxc3976429.pdf

Gowrishankar, S. Basavaraju, T.G. Manjaiah, D.H. & Subir, K.S. 2008. *Proceedings of the World Congress on Engineering, 2008 Vol. I, WCE 2008.* [online pdf]. Issues in Wireless Sensor Networks. [accessed 15 November 2011].

Available from: http://www.iaeng.org/publication/WCE2008/WCE2008_pp176-187.pdf

Holger, K. & Andreas, W. 2005. *Protocols and Architectures for Wireless Sensor Networks. Wiley*

Jangra. Swati. Richa & Priyanka 2010*. International Journal on Computer Science and Engineering (IJCSE).* [online pdf]. Wireless Sensor Network (WSN): Architectural Design Issues and Challenges. [accessed 15 November 2011].
Available from: http://www.enggjournals.com/ijcse/doc/IJCSE10-02-09-076.pdf

Jeremy, E. & Kay, R. 2002. *A New Regime for Time Synchronization. Princeton, New Jersey* [online pdf]. Wireless Sensor Networks. [accessed 15 November 2011].
Available from: http://www.vs.inf.ethz.ch/publ/papers/wsn-timesync.pdf

Kiyani, F. Aghaee_rad, A. & Tahmasebi_rad, H. 2011. *International Journal of Computer Science and Network Security.* [online pdf]. *DATS: A Distributed Algorithm for Time Synchronization in Wireless Sensor Network.* [accessed 20 December 2011]. Available from: http://paper.ijcsns.org/07_book/201102/20110235.pdf

Meinberg NTP Software, 2008. *Solution for Time and Frequency Synchronization* [web publication]. NTP Download [accessed 23 February 2012]. Available from: http://www.meinberg.de/english/info/ntp.htm

Marco, P.D. 2008. *Sweden: KTH Electrical Engineering. Master Degree Project.* [online publication]. *Protocol Design and Implementation for Wireless Sensor Networks.* [accessed on 16 November 2011].
Available from: http://www.ee.kth.se/~pidm/publications/thesis.pdf

National Instruments 2010. *Wireless Communications in WSN. NI WSN Link Quality Logger* [web publication]. NI Developer Zone [accessed 23 February 2012]. Available from: http://zone.ni.com/devzone/cda/epd/p/id/6346

Rahamatkar, S. Agarwal, A & Kumar, N. 2010. *International Journal on Computer Science and Engineering Vol. 02, No. 03, 2010.* [online journal]. Analysis and

Comparative Study of Clock Synchronization Schemes in Wireless Sensor Networks. [accessed 15 November 2011].
Available from: http://www.enggjournals.com/ijcse/doc/IJCSE10-02-03-28.pdf

Ranganathan, P. & Nygard, K. 2010. *International Journal of UbiCom (IJU), Vol.1, No.2.* [online journal]. Time Synchronization in Wireless Sensor Networks: A Survey. [accessed 10 February 2012].
Available from: http://airccse.org/journal/iju/papers/0410iju6.pdf

Roving Networks, 2011 *WiFly Evaluation Kit*. [online pdf]. User Manual. [accessed on 18 February 2012]. Available from: http://www.rovingnetworks.com/files/resources/rn-wifly-eval-um-ver1.0r.pdf

Roving Networks, 2011. *User Manual and Command Reference.* [online pdf]. 802.11 b/g Wireless LAN Modules.[accessed 18 February 2012]. Available from: http://www.rovingnetworks.com/products/RN_131G_EVAL

Sharma, S. 2009. *National Institute of Technology Rourkela, Department of Computer Science and Engineering. Master of Technology.* [Online pdf]. Energy - efficient Secure Routing in Wireless Sensor Networks. [accessed 25 October 2011].
Available from: http://ethesis.nitrkl.ac.in/1515/1/IOCMLB_thesis.pdf

Singh, S.K. Singh, M.P. & Singh, D.K. 2010. *International Journal of Computer Science & Engineering Survey (IJCSES) Vol. 1, No.2*. [online journal] Routing Protocols in Wireless Sensor Networks – A Survey. [accessed 08 November 2011].
Available from: http://airccse.org/journal/ijcses/papers/1110ijcses06.pdf

Sivrikaya, F. & Yener, B. 2004. *Time Synchronization in Sensor Networks: A Survey.* [online pdf]. [accessed 17 November 2011].
Available from: http://www.cs.rpi.edu/~yener/PAPERS/WINET/timesync04.pdf

The NTP FAQ and HOWTO, 2006. Understanding and Using the Network Time Protocol [web publication]. NTP [accessed 23 February 2012].
Available from: http://www.ntp.org/ntpfaq/NTP-s-algo.htm#Q-REFCLK

Viskström, P. 2006. *Time Stamping and Time Synchronization in Wireless Sensor Networks.* [online pdf]. [accessed 07 February 2012].
Available from: http://users.tkk.fi/virranko/sensor_networks/vikstrom1.pdf

APPENDICES

Appendix 1 Configuration file

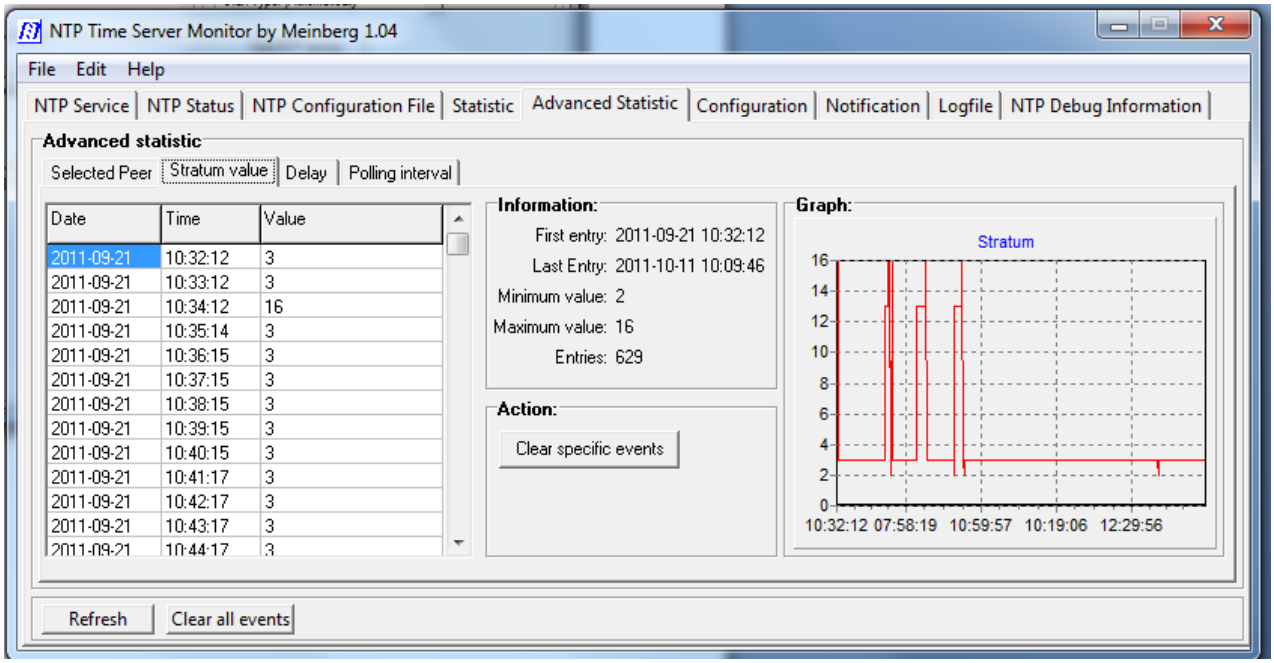
```
# NTP Network Time Protocol
# **** ATTENTION ****: *You have to restart the NTP service when you change this file
        to activate the changes*
# PLEASE CHECK THIS FILE CAREFULLY AND MODIFY IT IF REQUIRED
# Configuration File created by Windows Binary Distribution Installer Rev.: 1.26  mbg
# please check http://www.ntp.org for additional documentation and background
        information
# Use drift file
# your local system clock, should be used as a backup
# (this is only useful if you need to distribute time no matter how good or bad it is)
server 127.127.1.1
# but it operates at a high stratum level to let the clients know and force them to
# use any other timesource they may have.
fudge 127.127.1.1 stratum 12
        •
driftfile /etc/ntp.drift
driftfile "C:\Program Files\NTP\etc\ntp.drift"
broadcastdelay  0.008
broadcastclient
multicastmultcast
broadcast 192.168.3.255 ttl 6
broacast 224.0.0.1


# Use a NTP server from the ntp pool project (see http://www.pool.ntp.org)
server ns1.savonia.fi
server ntp.a-lab.fi iburst prefer

keys / etc/ inet/ ntp.keys
server ns1.savonia.fi prefer key 1
server ntp.a-lab.fi key 1
enable auth
trustedkey 1 1

# allow localhost
restrict 127.0.0.1
restrict 192.168.3.189

 Please note that you need at least four different servers to be at least protected against
peer 192.168.3.189
peer 192.168.3.100/24
```

keys / etc/ inet/ ntp.keys
trustedkeys 2 1
peer 192.168.3.189 key 2
peer 192.168.3.100 key 1

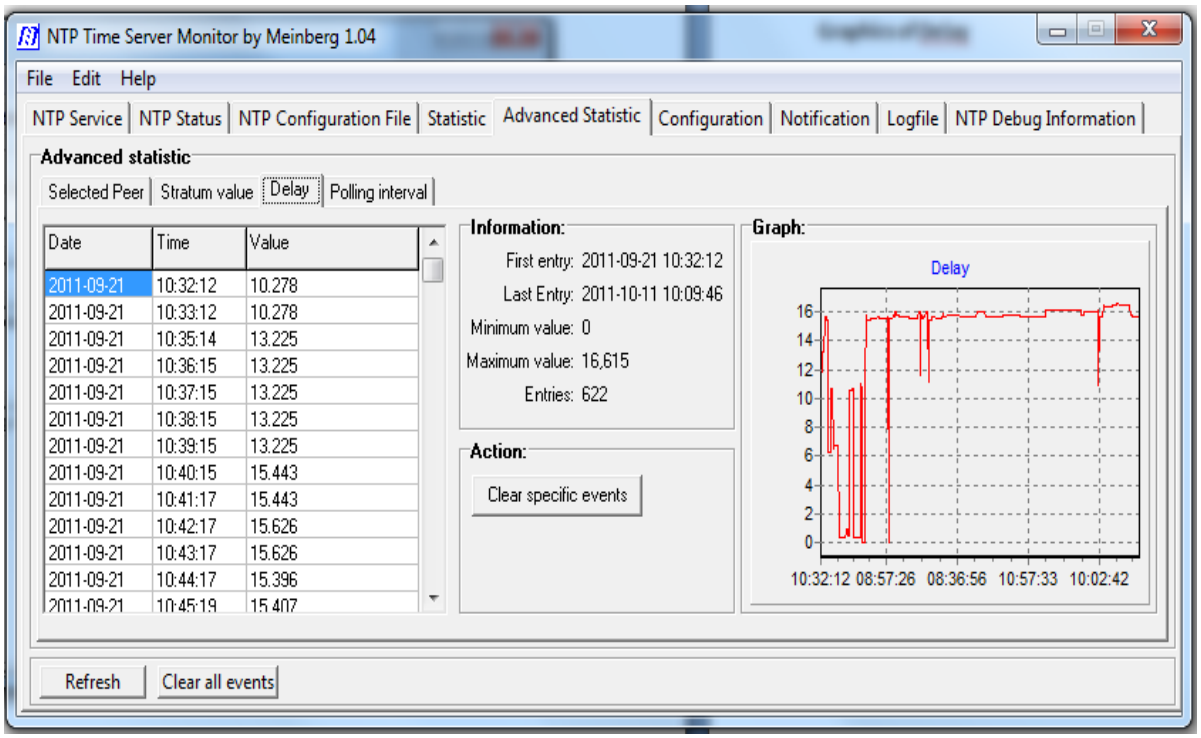
#accept packets from
restrict 192.168.3.100 mask 255.255.255.0 nomodify notrap
restrict 192.168.3.101 mask 255.255.255.0 nomodify notrap
restrict 192.168.3.189 mask 255.255.255.255 nomodify notrap noquery
restrict 192.168.3.100 mask 255.255.255.255 nomodify notrap noquery

one falseticker. If you only rely on internet time, it is highly recommended to add
additional servers here.
The 'iburst' keyword speeds up initial synchronization, please check the documentation
for more details!
server 0.fi.pool.ntp.org iburst
server 1.fi.pool.ntp.org iburst
#server 2.fi.pool.ntp.org iburst
#server 1.se.pool.ntp.org iburst
#server 0.de.pool.ntp.org iburst
End of generated ntp.conf --- Please edit this to suite your needs
##
#Section insert by NTP Time Server Monitor 30.8.2011
enable stats
statsdir "C:\Program Files\NTP\etc\"
statistics loopstats

Appendix 2 Graphical Display of Stratum Value



Appendix 3 Graphical Display of Delay



Appendix 4 Graphical Display of Polling Interval Delay